

REMARKS

Claims 10 and 21 are amended, and claims 25-33 are added herein. Claims 10-16 and 21-33 are pending in the captioned case. Reexamination and reconsideration of the presently claimed application are respectfully requested.

Allowable Subject Matter

Applicants appreciate the Examiner's indication that claims 16 and 24 contain allowable subject matter. However, in view of the amendments and remarks herein, Applicants contend that base claims 10 and 21, as well as added independent claim 25, are patentable over the cited references.

Objection to the Specification

An objection was lodged against the specification for an alleged informality. Specifically, the Examiner believes there is an embedded hyperlink in the specification at page 6, line 14. Applicants respectfully disagree. The purported hyperlink is instead merely the full web address (in textual form) for the cited article. There is no embedded hyperlink or browser-executable code in the present specification. Accordingly, Applicants respectfully request removal of this objection.

Section 103 Rejection

Claims 10-15 and 21-23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,124,405 to Kakivaya et al. (hereinafter "Kakivaya"), U.S. Patent No. 5,764,976 to Hsiao (hereinafter "Hsiao"), and U.S. Patent No. 5,999,729 to Tabloski (hereinafter "Tabloski"). It is believed the amendments herein more clearly identify the differences between the present invention and that of Kakivaya, Tabloski, and Hsiao, both singularly and in combination.

There are several points of agreement with the Office Action of June 2, 2009. First, Applicant agrees that “Kakivaya did not specifically teach determining if a deadlock exists by building a wait graph of the blocked threads” (Office Action -- pg. 3, ¶ 8). However, the Office Action posits that Hsiao teaches building a wait graph of the blocked threads (Office Action -- pg. 3, ¶ 9). While Hsiao does build a wait graph (Hsiao at col. 3), it does so in a completely different setting than the present application. In Hsiao, a relational database is built with table entries for a “locked” state.

Hsiao does not relate to a multi-thread system, dataflow applications, composite ports and components, allocation of thread to deadlock detection, and other features found in the present claims. It is not seen how the modification can be made to Kakivaya. The Office Action acknowledges that Kakivaya and Hsiao are deficient in these and other respects (Office Action -- pg. 4, ¶ 11).

Kakivaya is directed to automatic initialization and class initialization methods which determines behavior in both single and multithreaded environments. Kakivaya includes deadlock detection as part of execution of the initialization code. As such, Kakivaya is primarily concerned with class dependency loops (Kakivaya at col. 1, lines 57-65). Kakivaya does build a graph for such class dependencies.

Tabloski describes a deadlock detection mechanism, but Tabloski does not monitor deadlock on a thread level. Tabloski does not execute map processes on a separate thread. Instead, Tabloski executes objects concurrently as separate processes – not in a shared memory architecture. There are no composite components or composite maps. Therefore, it is not surprising that in Tabloski, the execution objects communicate blockage to an execution control object (a processor).

Claim 1 is amended to reflect that a thread is used in the present deadlock monitoring method. Support can be found for the amendments to claim 1 in the present specification, for example, on pg. 30, lines 1-15 and pgs. 32-34. Of course, block detection does not have to be a dedicated task for a block detection thread, but can be. The block detection thread monitors the data

queues and does not require communication from the operators (*see*, added claims 32-33). Claim 21 of the present invention explains how a data queue may be expanded as necessary based on information collected in the wait graph. Tabloski does not use data queues or block detection threads as part of its deadlock detection and management, nor can Tabloski be modified to do so.

Tabloski describes a different type of deadlock detection mechanism for managing deadlock across multiple processes. Tabloski uses an execution control object on each node of the networked processing environment to communicate with the execution objects. Amended claim 10 does not use a control object. Tabloski does not monitor deadlock detection on a thread level. The execution objects of Tabloski communicate blockage to an execution control object. In the present deadlock monitoring invention (e.g., claim 10), the deadlock monitoring thread watches the data queues (*see* claim 32) and does not require communication from the operators as required in Tabloski. In claim 21, the present invention expands the queues if necessary based on information it has collected in the wait graph.

A key advantage to the use of a deadlock detection method of the present invention is to provide an automatic mechanism for dealing with different rates of flow of the tokens through the dataflow graph. Some operators need more time to process data than others or may need to aggregate data, so that operators consume data at a different rate than they emit it. The queues of the present invention allow for multiple downstream consumers on the same stream of data. Tabloski does not use queues, so outbound ports in Tabloski buffer data. The buffers of Tabloski are not analogous to the data queues in the present application. Further, Tabloski appears to be a conventional multiprocessor approach and is primarily concerned with getting data to all processors.

Claims 25-31 are added to emphasize features of the present invention which are impossible to replicate with Tabloski and any combination with Kakivaya or Hsiao. Support for added claim 25 can be found in the present specification, for example, pages 30-34.

For at least the reasons stated above, Applicants believe present claims 10-16 and 21-24, as well as added claims 25-33, are patentably distinct over the cited references. Accordingly, removal of this rejection is respectfully requested.

CONCLUSION

The present amendment and response is believed to be a complete response to the issues raised in the Office Action mailed June 2, 2009. In view of the amendments and remarks herein, Applicants assert pending claims 10-16 and 21-33 are in condition for allowance. If the Examiner has any questions, comments, or suggestions, the undersigned attorney earnestly requests a telephone conference.

No fees are required for filing this amendment; however, the Commissioner is authorized to charge any additional fees which may be required, or credit any overpayment, to Daffer McDaniel, LLP Deposit Account No. 50-3268.

Respectfully submitted,

/Charles D. Huston/

Charles D. Huston

Reg. No. 31,027

Attorney for Applicant(s)

Customer No. 35617

Date: October 22, 2009